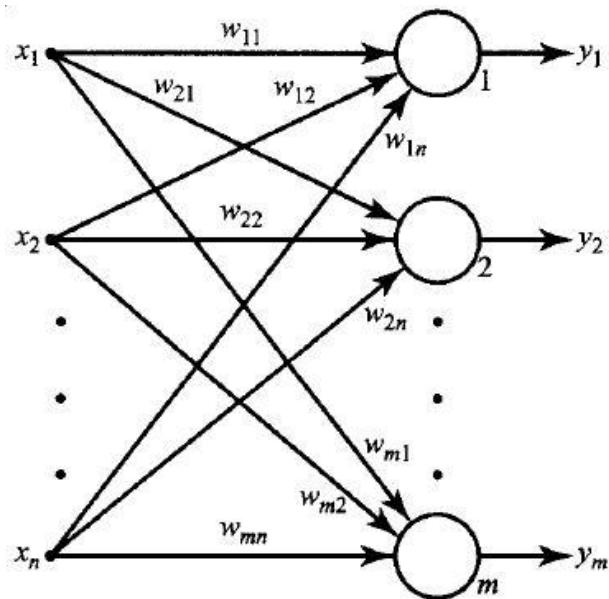


„Podstawy Sztucznej Inteligencji”

Scenariusz 2

Temat ćwiczenia: *Budowa i działanie sieci jednowarstwowej*

1. Sieć jednowarstwowa – tworzą ją neurony ułożone w jednej warstwie. Każdy z neuronów posiada polaryzację oraz wiele wag prowadzących od sygnałów wejściowych. Wagi są dobierane w procesie uczenia.



1. Wykorzystany algorytm uczenia (dla liter Aa, Ee, Hh, Ii, Ll, Oo, Pp, Tt, Uu, Zz)

Algorytm:

```
close all; clear all; clc;
```

```
Wejscie=[0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;
          0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1; 0 1;]
```

```
Wyjscie=1;
```

```
net = newlin(Wejscie,Wyjscie,0,0.01);
```

```
%A a E e H h I i L l O o P p T t U u Z z
WE=[0 0 1 1 1 1 1 1 1 1 1 0 0 1 0 1 0 1 0 1 0
     1 0 1 1 0 0 0 0 0 0 0 1 0 1 0 1 1 0 0 1 0
     1 0 1 1 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 1 0
     0 0 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0
     1 0 1 1 1 1 1 0 1 1 1 0 1 1 0 1 1 0 0 1
     0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 0 1
     0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 1
     1 0 0 1 1 0 0 0 0 0 0 1 0 1 0 0 0 1 0 1 1
     1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 0
     1 1 1 1 1 1 0 0 0 0 0 0 1 1 0 1 0 0 0 0]
```

```

1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 0 0 1 1 1
1 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 1 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0
1 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 0 0 0
1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 0 1 1 1 1
0 1 1 1 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1
0 1 1 1 0 0 0 0 1 1 1 1 0 0 1 1 1 1 1 1
1 1 1 1 1 1 0 0 1 0 0 0 0 0 0 1 1 1 1 1

```

```
WY=[1 0 1 0 1 0 1 0 1 0 1 0];
```

```
net.name='Wielkosc litery';
```

```
net.trainParam.epochs = 2500;
net.trainParam.goal = 0.001;
net.trainParam.mu = 0.001;
```

```
net = train(net, WE, WY);
```

```
Y=sim(net, WE);
```

```
testA= [0;1;1;0;
        1;0;0;1;
        1;1;1;1;
        1;0;0;1;
        1;0;0;1;];
```

```
testa= [0;0;0;0;
        0;0;0;0;
        1;1;1;0;
        1;0;1;0;
        1;1;1;1;];
```

```
testE= [1;1;1;1;
        1;0;0;0;
        1;1;1;1;
        1;0;0;0;
        1;1;1;1;];
```

```
teste= [1;1;1;1;
        1;0;0;1;
        1;1;1;1;
        1;0;0;0;
        1;1;1;1;];
```

```
testH= [1;0;0;1;
        1;0;0;1;
        1;1;1;1;
        1;0;0;1;
        1;0;0;1;];
```

```
testh= [1;0;0;0;
        1;0;0;0;
        1;1;1;1;
        1;0;0;1;
        1;0;0;1;];
```

```
testI= [1;0;0;0;
        1;0;0;0;
        1;0;0;0;
        1;0;0;0;
        1;0;0;0;];
```

```
testi= [1;0;0;0;
        0;0;0;0;];
```

```
1;0;0;0;
1;0;0;0;
1;0;0;0];

testL= [1;0;0;0;
1;0;0;0;
1;0;0;0;
1;0;0;0;
1;1;1;1];
testl= [1;0;0;0;
1;0;0;0;
1;0;0;0;
1;0;0;0;
1;1;1;0];

testO= [0;1;1;0;
1;0;0;1;
1;0;0;1;
1;0;0;1;
0;1;1;0;];
testo= [0;0;0;0;
0;1;1;0;
1;0;0;1;
1;0;0;1;
0;1;1;0;];

testP= [1;1;1;1;
1;0;0;1;
1;1;1;1;
1;0;0;0;
1;0;0;0;];
testP= [0;0;0;0;
1;1;1;0;
1;1;1;0;
1;0;0;0;
1;0;0;0;];

testT= [1;1;1;1;
0;0;1;0;
0;0;1;0;
0;0;1;0;
0;0;1;0;];
testt= [0;1;0;0;
1;1;1;0;
0;1;0;0;
0;1;0;0;
0;1;1;0;];

testU= [1;0;0;1;
1;0;0;1;
1;0;0;1;
1;0;0;1;
1;1;1;1;];
testu= [0;0;0;0;
0;0;0;0;
1;0;1;0;
1;0;1;0;
1;1;1;1;];

testZ= [1;1;1;1;
0;0;0;1;
0;0;1;0;
0;1;0;0;
1;1;1;1;];
```

```

testz= [0;0;0;0;
        1;1;1;1;
        0;0;1;0;
        0;1;0;0;
        1;1;1;1;];

efekt=sim(net, testa);

efekt

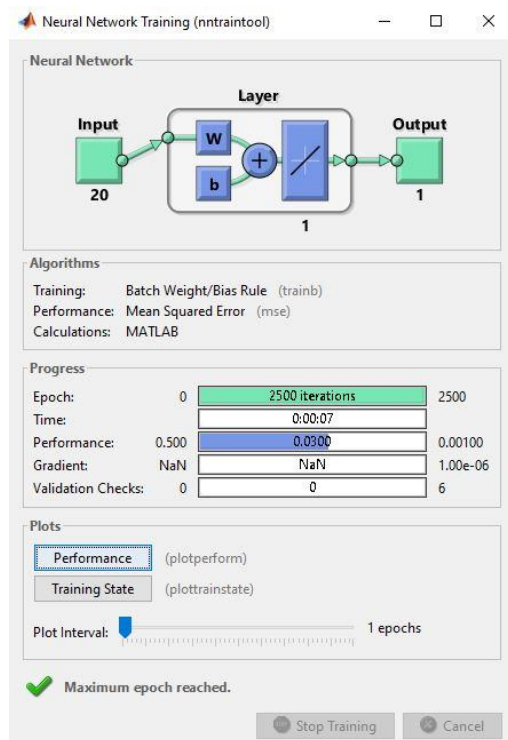
```

Funkcja:

- `newp` tworzy jednowarstwową sieć neuronową.
- `sim` symuluje działanie perceptronu.
- `test` wyświetla dane testowe
- `efekt` wyświetla wyniki które przewidzi perceptron dla powyższych danych
- `net` to struktura, która zawiera opis sieci jednokierunkowej
- `trainParam.goal` jest to błąd średniokwadratowy
- `trainParam.mu` jest to współczynnik uczenia

2. Otrzymane wyniki

Rysunek 1



Rysunek 2

[illegible]

3. Wnioski

Do nauczenia perceptron potrzebuje 2500 iteracji.

Na rysunku 2 efekt wynosi w przybliżeniu 0 stąd wynika że litera jest literą małą.